IN THE CLAIMS

1. (original) A system comprising:

a system bus;

a direct memory access (DMA) controller coupled to the system bus;

a memory control device coupled to the system bus and the DMA controller, the memory control device including a memory controller, a memory read aligner, a memory write aligner and a memory slave interface;

a main memory coupled to the memory control device; and

a slave device coupled to the system bus and the DMA controller, the slave device including a slave peripheral, a peripheral read aligner, a peripheral write aligner and a peripheral slave interface.

2. (original) The system of Claim 1, wherein the DMA controller is configured to implement fly-by read and fly-by write operations between the memory control device and the slave device.

3. (original) The system of Claim 2, wherein the memory read aligner is configured to provide data words that are fully aligned with the system bus during fly-by write operations.

4. (original) The system of Claim 2, wherein the peripheral read aligner is configured to provide data words that are fully aligned with the system bus during fly-by read operations.

SN: 09/888,321

5. (original) The system of Claim 2, wherein the memory write aligner is configured to process data words that are fully aligned with the system bus during fly-by read operations.

6. (original) The system of Claim 5, wherein the memory write aligner is further configured to re-align the data words on the system bus to a byte address of the main memory during fly-by read operations.

7. (original) The system of Claim 2, wherein the peripheral write aligner is configured to process data words that are fully aligned with the system bus during fly-by write operations.

8. (original) The system of Claim 7, wherein the peripheral write aligner is further configured to re-align the data words on the system bus to a byte address of the slave peripheral during fly-by write operations.

9. (original) The system of Claim 2, wherein the slave peripheral includes a first in, first out (FIFO) memory coupled to the peripheral read aligner and the peripheral write aligner.

10. (original) The system of Claim 9, wherein the slave device comprises means for adjusting a read pointer of the FIFO memory after a fly-by read operation.

11. (original) The system of Claim 9, wherein the slave peripheral further comprises means for informing the peripheral read aligner and the peripheral write aligner of

an alignment offset between the system bus and the FIFO memory.

12. (original) The system of Claim 1, wherein each of the aligners include:

a barrel shifter coupled to receive a data word, a byte enable value and an alignment signal;

a register coupled to receive a data word and a byte enable value provided by the barrel shifter; and

an output multiplexer coupled to receive the data words and byte enable values provided by the barrel shifter and the register, the output multiplexer being controlled by the alignment signal.

13. (original) The system of Claim 12, wherein each register includes:

a data register for storing a data word provided by the barrel shifter;

a byte enable register for storing a byte enable value provided by the barrel shifter; and

means for clearing the byte enable register.

14. (original) The system of Claim 12, wherein each aligner further includes:

means for preloading the register with a data word and byte enable value provided by the barrel shifter;

means for loading the register with a data word and byte enable value provided by the barrel shifter;

means for preserving a data word and a byte enable value in the register; and

means for bypassing the barrel shifter and the register.

SN: 09/888,321

15. (original) A method of performing a fly-by read operation, the method comprising the steps of:

reading data words from a memory device in a slave peripheral, wherein each of the data words includes N bytes;

incrementing a read pointer of the memory device each time a data word is read from the memory device;

aligning the data words read from the memory device to a system bus; and

adjusting the read pointer at the end of the fly-by read operation.

16. (original) The method of Claim 15, further comprising the step of providing byte enable signals on the system bus using a direct memory access (DMA) controller, the byte enable signals corresponding with bytes of the data words being transferred.

17. (original) The method of Claim 16, wherein the step of adjusting the read pointer comprises the steps of:

backing up the read pointer by one if the data words read from the memory device are initially aligned with the system bus and a last byte enable signal of the fly-by read operation indicates that all of the bytes of a last data word are valid; and

backing up the read pointer by two if the data words read from the memory device are initially aligned with the system bus and the last byte enable signal indicates that not all of the bytes of the last data word are valid.

18. (original) The method of Claim 16, wherein the step of adjusting the read pointer comprises the steps of:

backing up the read pointer by one if the data words read from the memory device must be shifted by one byte to become aligned with the system bus and a last byte enable signal of the fly-by read operation indicates that at least N-1 bytes of a last data word are valid; and

backing up the read pointer by two if the data words read from the memory device must be shifted by one byte to become aligned with the system bus and the last byte enable signal indicates less than N-1 bytes of the last data word are valid.

19. (original) The method of Claim 16, wherein the step of adjusting the read pointer comprises the steps of:

backing up the read pointer by one if the data words read from the memory device must be shifted by two bytes to become aligned with the system bus and a last byte enable signal of the fly-by read operation indicates that at least N-2 bytes of a last data word are valid; and

backing up the read pointer by two if the data words read from the memory device must be shifted by two bytes to become aligned with the system bus and the last byte enable signal indicates that less then N-2 bytes of the last data word are valid.

20. (original) The method of Claim 16, wherein the step of adjusting the read pointer comprises the step of backing up the read pointer by one if the data words read

from the memory device must be shifted by N-1 bytes to become aligned with the system bus.

21. (original) The method of Claim 16, further comprising the steps of:

providing an alignment signal on the system bus using the direct memory access (DMA) controller; and

re-aligning data words from the system bus to a byte location in a main memory in response to the alignment signal.

22. (original) The method of Claim 21, further comprising the step of providing a last address and a byte enable signal having a logic zero value using the DMA controller at the end of the fly-by read operation.

23. (original) The method of Claim 15, further comprising the step of operating the memory device in a first in, first out (FIFO) manner.

24. (original) A method of performing a fly-by write operation between a memory device and a slave device on a system bus, the method comprising:

reading a first data word from the memory device, wherein the first data word is not aligned with the system bus, such that the first data word includes a first set of one or more bytes to be included in the fly-by write operation, and a second set of one or more bytes to be excluded from the fly-by write operation;

transmitting an invalid data word on the system bus to a slave device, wherein the invalid data word

includes the second set of one or more bytes of the first data word;

transmitting a disabling byte-enable value on the system bus to the slave device, the disabling byte-enable value corresponding with the invalid data word;

preventing the invalid data word from being written to the slave device in response to the disabling byte-enable value.

25. (original) The method of Claim 24, further comprising providing the disabling byte-enable value with a direct memory access (DMA) controller.

26. (original) The method of Claim 25, further comprising:

providing a first address signal from the DMA controller to the memory device; and

retrieving the first data value from the memory device in response to first address signal.

27. (original) The method of Claim 24, further comprising:

reading a second data word from the memory device, wherein the second data word is not aligned with the system bus;

combining the first set of one or more bytes of the first data word with a first set of one or more bytes of the second data word to create a first fly-by write data word;

transmitting the first fly-by write data word on the system bus to the slave device;

transmitting a first byte-enable word on the system bus to the slave device, the first byte-enable word corresponding with the first fly-by write data word; and

writing the first fly-by write data word to the slave device in response to the first byte-enable word.

28. (original) The method of Claim 27, further comprising shifting the first fly-by write data word in the slave device in response to an alignment signal.

29. (original) The method of Claim 27, further comprising:

providing an alignment signal that identifies a degree of misalignment between the first data word and the system bus; and

combining the first set of one or more bytes of the first data word with a first set of one or more bytes of the second data word in response to the alignment signal.

30. (original) The method of Claim 29, further comprising providing the alignment signal to the memory device with a direct memory access (DMA) controller.

31. (original) The method of Claim 24, further comprising:

providing a Ready signal on the system bus when the memory device is in a Ready state; and

providing a Wait signal on the system bus when the memory device is in a Wait state.

32. (original) The method of Claim 24, further comprising the step of operating the slave device in a first in, first out (FIFO) manner.

A1

33. (original) A method of performing a fly-by read operation, the method comprising:

reading a first data word from a first memory device;

preloading the first data word into a read aligner; and then

reading a second data word from the first memory device;

applying the second data word to the read aligner;

applying a first alignment signal to the read aligner;

creating a first fly-by read word by routing a first portion of the first data word and a first portion of the second data word through the read aligner in response to the first alignment signal; and

transmitting the first fly-by read word on a system bus to a write aligner.

34. (original) The method of Claim 33, further comprising:

loading the second data word into the read aligner after the first fly-by read word is transmitted to the system bus;

reading a third data word from the first memory device;

applying the third data word to the read aligner;

creating a second fly-by read word by routing a second portion of the second data word and a first

portion of the third data word through the read aligner in response to the first alignment signal; and

transmitting the second fly-by read word on a system bus to the write aligner.

35. (original) The method of Claim 33, further comprising:

applying a second alignment signal to the write aligner; and

routing a first portion of the first fly-by read word through the write aligner in response to the second alignment signal;

providing a first byte enable signal in response to the second alignment signal;

applying the first portion of the first fly-by read word and the first byte-enable signal to a second memory device; and

writing the first portion of the first fly-by read word to the second memory device, wherein the first byte enable signal enables the first portion of the first fly-by read word to be written to the second memory device.

36. (original) The method of Claim 35, further comprising:

providing a Ready signal on the system bus when the second memory device is in a Ready state; and

providing a Wait signal on the system bus when the second memory device is in a Wait state.